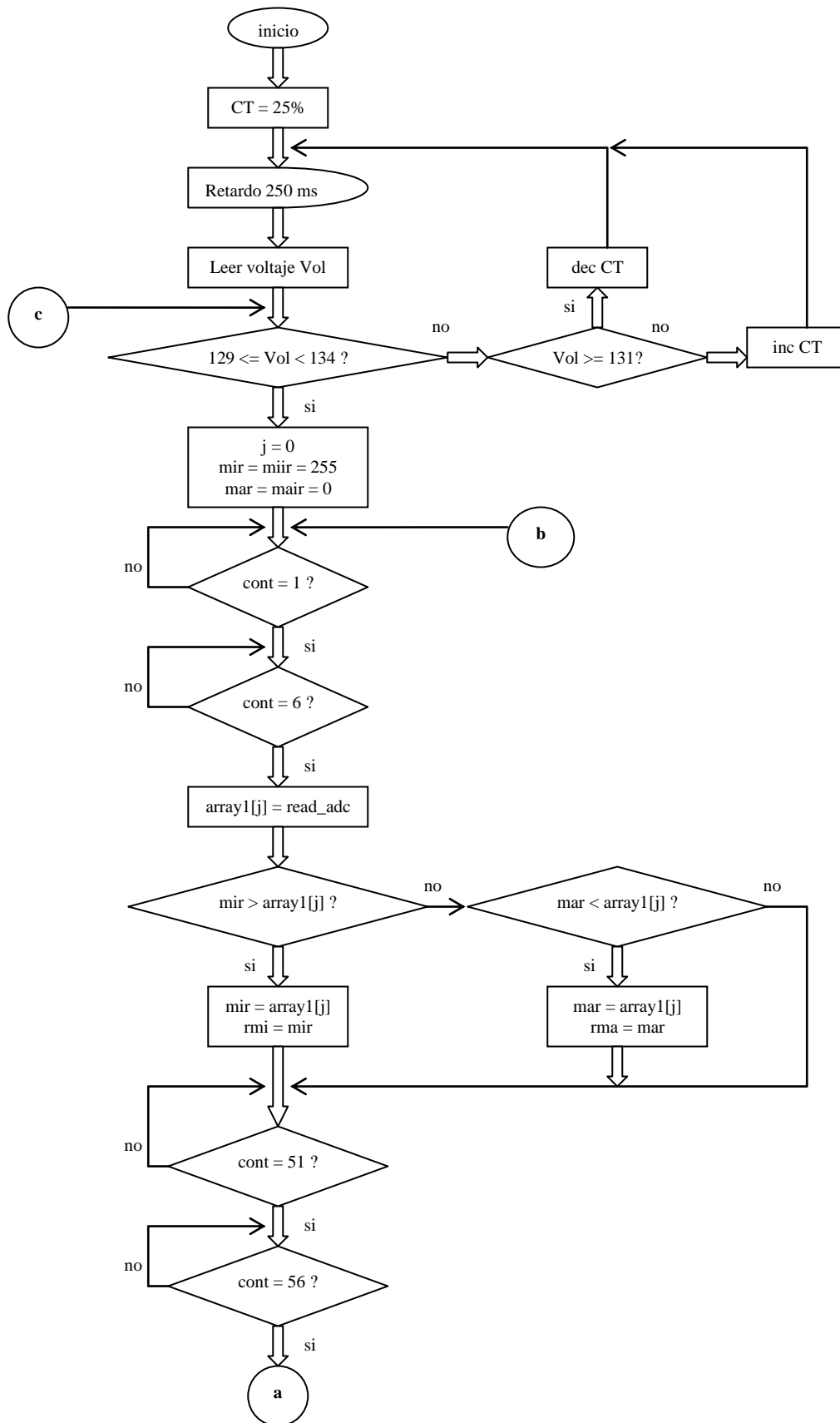
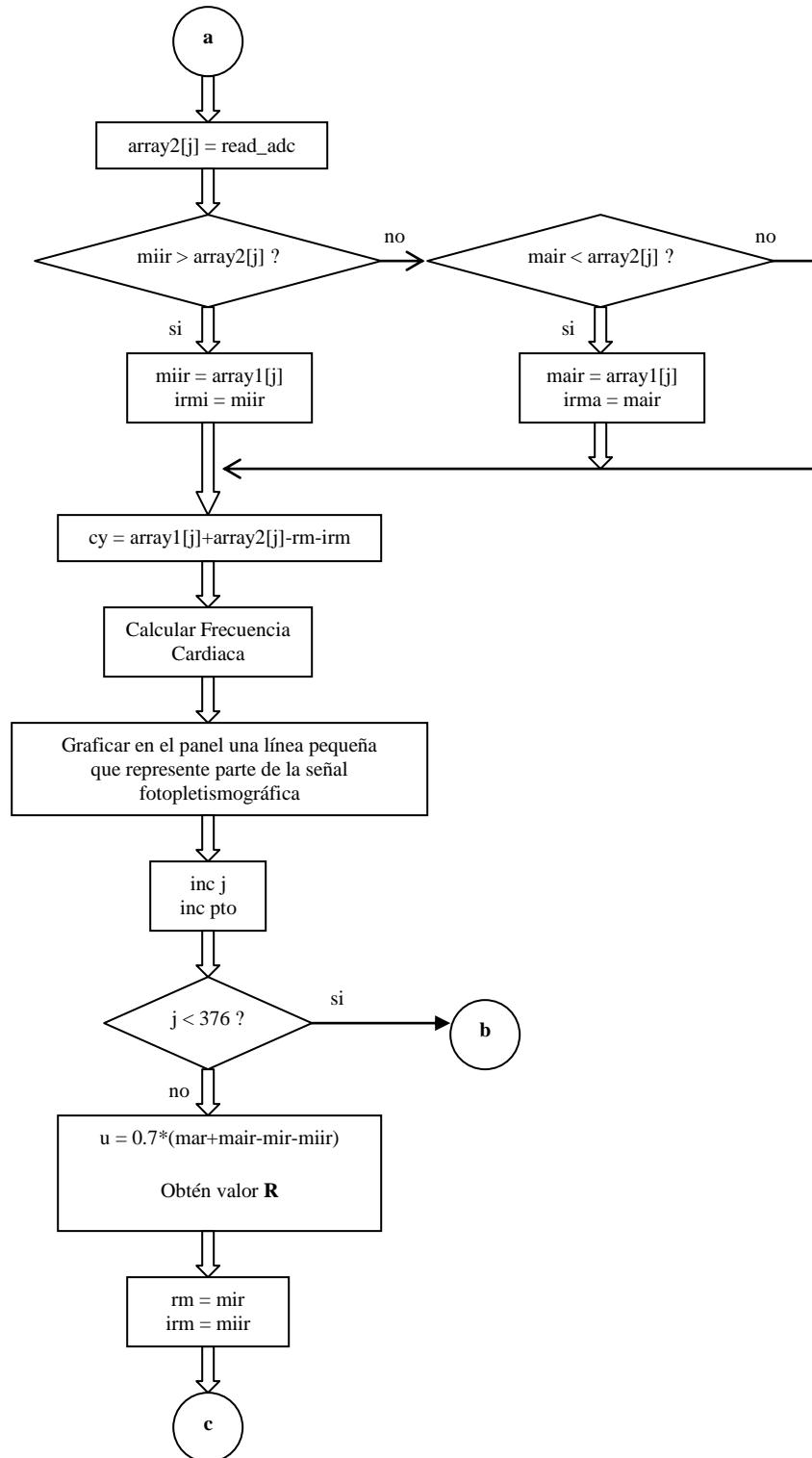


# ANEXOS

## DIAGRAMA DE FLUJO PRINCIPAL



Continuación del diagrama de flujo.





```

        SpO2=71; return(SpO2);}
        SpO2=72; return(SpO2);}
        SpO2=73; return(SpO2);}
        SpO2=74; return(SpO2);}
        SpO2=75; return(SpO2);}
        SpO2=76; return(SpO2);}
        SpO2=77; return(SpO2);}
        SpO2=78; return(SpO2);}
        SpO2=79; return(SpO2);}
        SpO2=80; return(SpO2);}
        SpO2=81; return(SpO2);}
        SpO2=82; return(SpO2);}
        SpO2=83; return(SpO2);}
        SpO2=84; return(SpO2);}
        SpO2=85; return(SpO2);}
        SpO2=86; return(SpO2);}
        SpO2=87; return(SpO2);}
        SpO2=88; return(SpO2);}
        SpO2=89; return(SpO2);}
        SpO2=90; return(SpO2);}
        SpO2=91; return(SpO2);}
        SpO2=92; return(SpO2);}
        SpO2=93; return(SpO2);}
        SpO2=94; return(SpO2);}
        SpO2=95; return(SpO2);}
        SpO2=96; return(SpO2);}
        SpO2=97; return(SpO2);}
        SpO2=98; return(SpO2);}
        SpO2=99; return(SpO2);}
        SpO2=100; return(SpO2);}
    }

Calc_FC()          // Función para calcular la frecuencia cardiaca
{
    iFC: if (DJ == 1)      // iFC: inicia Frecuencia Cardiaca
        {if (c1 < 99)    // DJ: Dejar pasar
            {c1++;
             return(FC);}
          DJ = 0;
          c1 = 0;
          goto iFC;}
    if (cy > u)           // cy : coordenada y
        {F4 = 11480(100+c2); // FC: Frecuencia cardiaca.
         FC = (F0+F1+F2+F3+F4)/5;
         F0 = F1, F1 = F2, F2 = F3, F3 = F4;
         DJ = 1;
         c2 = 0;
         goto iFC;}
    c2++;
    return(0);
}

valorR()           // Función que devuelve el promedio de 5 valores R (1 valor actual y 4 anteriores)
{
    R4 = (rma-rmi)/(irma-irmi);
    R=(R0+R1+R2+R3+R4)/5;
    R0=R1;
    R1=R2;
    R2=R3;
    R3=R4;
    return(R);
}

#int_timer2
void timer2_isr() // Interrupción que entra cada 40 us (25000 interrupciones por segundo)
{                // Se utiliza PWM para cada LED, donde sus CT (Ciclo de Trabajo) comprenden valores de 0 a 50 %
}

```

```

cont++;
if (cont>=51)
{
if (cont<=CT+50)
{
output_high(LIR);
}
else
output_low(LIR);
if (cont==100)
{
cont=0;
}
else
break;
}
else
if (cont<=CT)
{
output_high(LR);
output_low(LIR);
}
else
output_low(LR);
}

void main()                // PROGRAMA PRINCIPAL
{
unsigned long j=0;         // j: Para determinar el número de elementos de cada arreglo.
unsigned int Ref;         // Ref: Sirve para leer los 2 V de referencia por canal A0.
unsigned int array1[375]; // Si se llega a tomar la frec. card. un atleta, su pulso mínimo normal en reposo sería de
                          // 40 lat/min, esto equivale a que su pulso dura 1.5 s, entonces se toman (375 muestras de cada arreglo)/(250 muestras/s)

unsigned int array2[375];
glcd_init(ON);
set_tris_e(0);
setup_adc(adc_clock_internal);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
setup_spi(FALSE);
set_timer0(0);
setup_timer_0(RTCC_INTERNAL);
set_timer2(0);
setup_timer_2(T2_DIV_BY_1,239,2); // 239 es para que esté activandose cada led a una frecuencia de 250 Hz.
                                  //TMR2 cuenta de 0 a 239 (240) 2 veces, entonces interrumpira cada 40 us.
                                  // 480/12MHz = 40 us/interrupción ==> 25000 interrupciones/s

enable_interrupts(INT_TIMER2);
enable_interrupts(GLOBAL);
output_low(LR);
SpO2 = 90;
cont = 0;
CT = 25; //A partir de un CT (Ciclo de Trabajo) de 6% (240 us) empieza a estabilizarse la respuesta del fotodiodo cuando está
         //activandose el led INFRA.
FC = 60; // FC : frecuencia cardiaca
FCA = 60; // FCA : frecuencia cardiaca anterior
DJ = 0; // DJ : dejar pasar
c1 = 0; // c1 : contador 1
c2 = 0; // c2 : contador 2
cy = 0;
cya = 0; // crdy : coordenada en y anterior
crdy = 0; // crdy : coordenada en y
u = 1; // u : umbral
R = 1;
R0 = 1;
R1 = 1;
R2 = 1;

```

```

R3 = 1;
R4 = 1;
rm = 0;
irm = 0;
rmi = 0;
rma = 0;
irmi = 0;
irma = 0;
pto = 0; // cpto : para enviar 1 punto de cada 6 muestras al LDC Gráfico
cx = 0; // cx : coodenada x actual del LCD Gráfico
cxa = 0; // cxa : coordenada x anterior del LCD Gráfico
while(true)
{
ini: delay_ms(250); //Tiempo necesario que espera el uC para preguntar si ha alcanzado los 2 Volts de Referencia.

ini1: set_adc_channel(0); //Canal A0 para un voltaje de control de 2.5V.
Ref = read_adc(); // Leé Vol: Volumen

if ((Ref>=129)&&(Ref<134)) // Pregúnta si Vol ó Ref está entre esos dos valores
{
j = 0;
mir = 255;
mar = 0;
miir = 255;
mair = 0;
preg: if (cont==1)
{
preg1: if (cont==6)
{
set_adc_channel(1); //Canal A1 para el muestreo del reflejo de la onda cuando emite el LED ROJO.
array1[j]=read_adc();

if (mir > array1[j])
{mir = array1[j];
rmi = mir;} // rmi: rojo mínimo
if (mar < array1[j])
{mar = array1[j];
rma = mar;} // rma: rojo máximo
preg2: if (cont==51)
{
preg3: if (cont==56)
{
array2[j]=read_adc(); //Canal A1 para el muestreo del reflejo de la onda cuando emite el LED INFRA.

if (miir > array2[j])
{miir = array2[j];
irmi = miir;} // irmi: ir mínimo
if (mair < array2[j])
{mair = array2[j];
irma = mair;} // irma: ir máximo

cy = array1[j]+array2[j]-rm-irm;
Calc_FC();
if (pto>4)
{
//output_high(SAL); // Señal para indicar el momento en que el LDCG va a trazar 43 líneas
crdy=63-cy; // cada segundo, para que sean (3*43=129) líneas que corresponden a los
glcd_line(cxa,cya,cx,crdy,ON); // 128 pixeles del display gráfico.
glcd_update();
pto=0;
cxa=cx;
cya=crdy;
cx++;
//output_low(SAL);

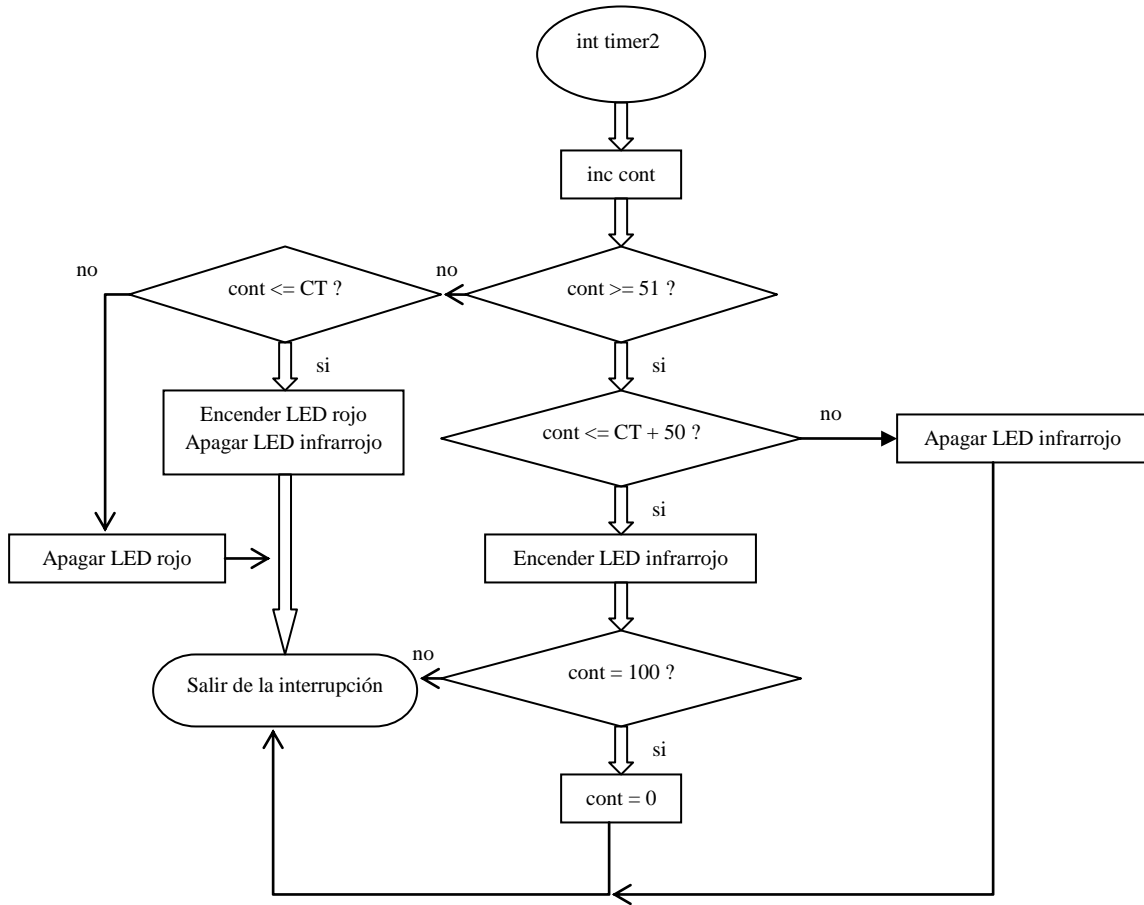
```

```

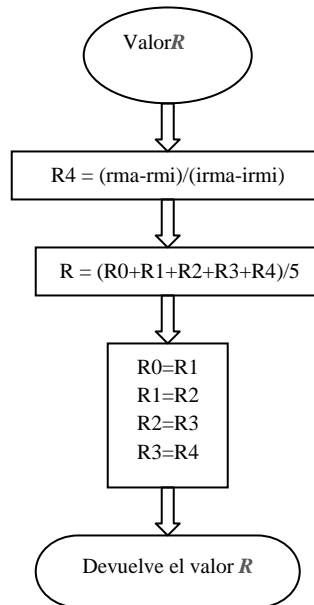
    if (cx>127)
    {
        glcd_fillScreen(OFF);
        cxa=0;
        cx=1;
        cya=crdy;
    }
    }
    pto++;
    j++;
if (j<376) // Frecuencia de muestreo = 250 muestras/s
    { // ó también (375 muestras) / (250 muestras/s) = 1.5 s
        goto preg; // (60 latidos/min)/(1.5s) = 40 latidos/min = 40 lpm
    } // corresponde a que el sistema puede registrar a partir de una frecuencia mínima de 40 lpm.
else
    u = 0.7*(mar+mair-mir-miir);
    valorR();
    Psat();
    dsplyV();
    rm = mir;
    irm = miir;
    goto ini1;
    }
    goto preg3;
    }
else
    goto preg2;
    }
else
    goto preg1;
    }
else
    goto preg;
    }
if (Ref>=131)
    {
        CT--;
        goto ini;
    }
else
    CT++;
    goto ini;
    }
}
}

```

## Diagrama de flujo para la interrupción del timer2

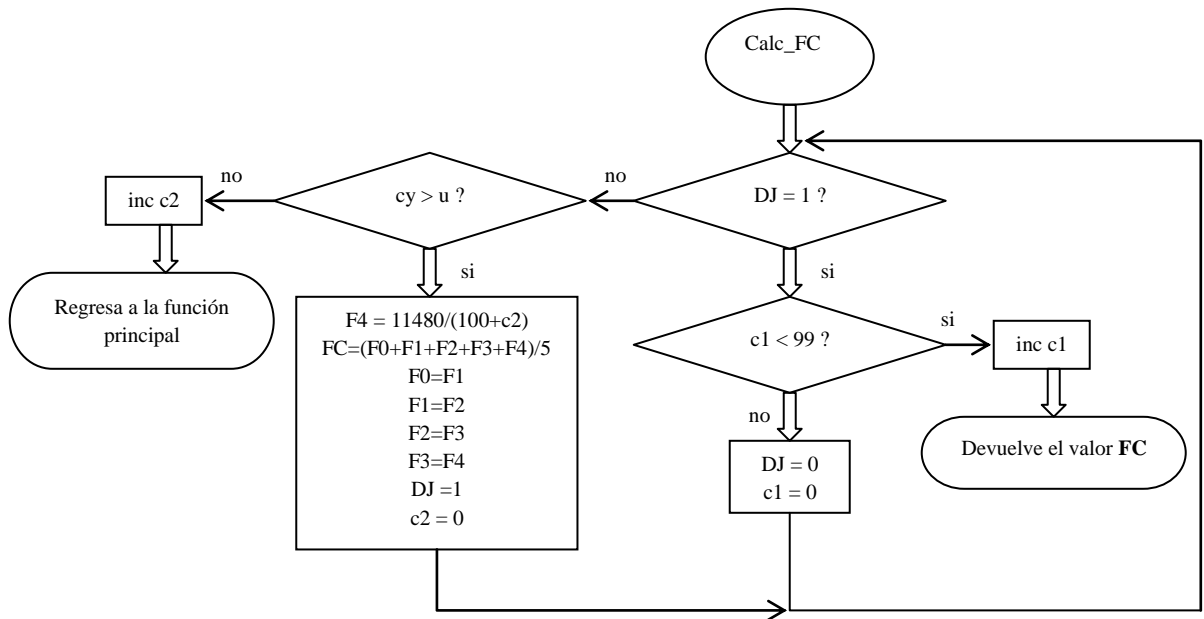


## Diagrama de flujo para calcular el valor R

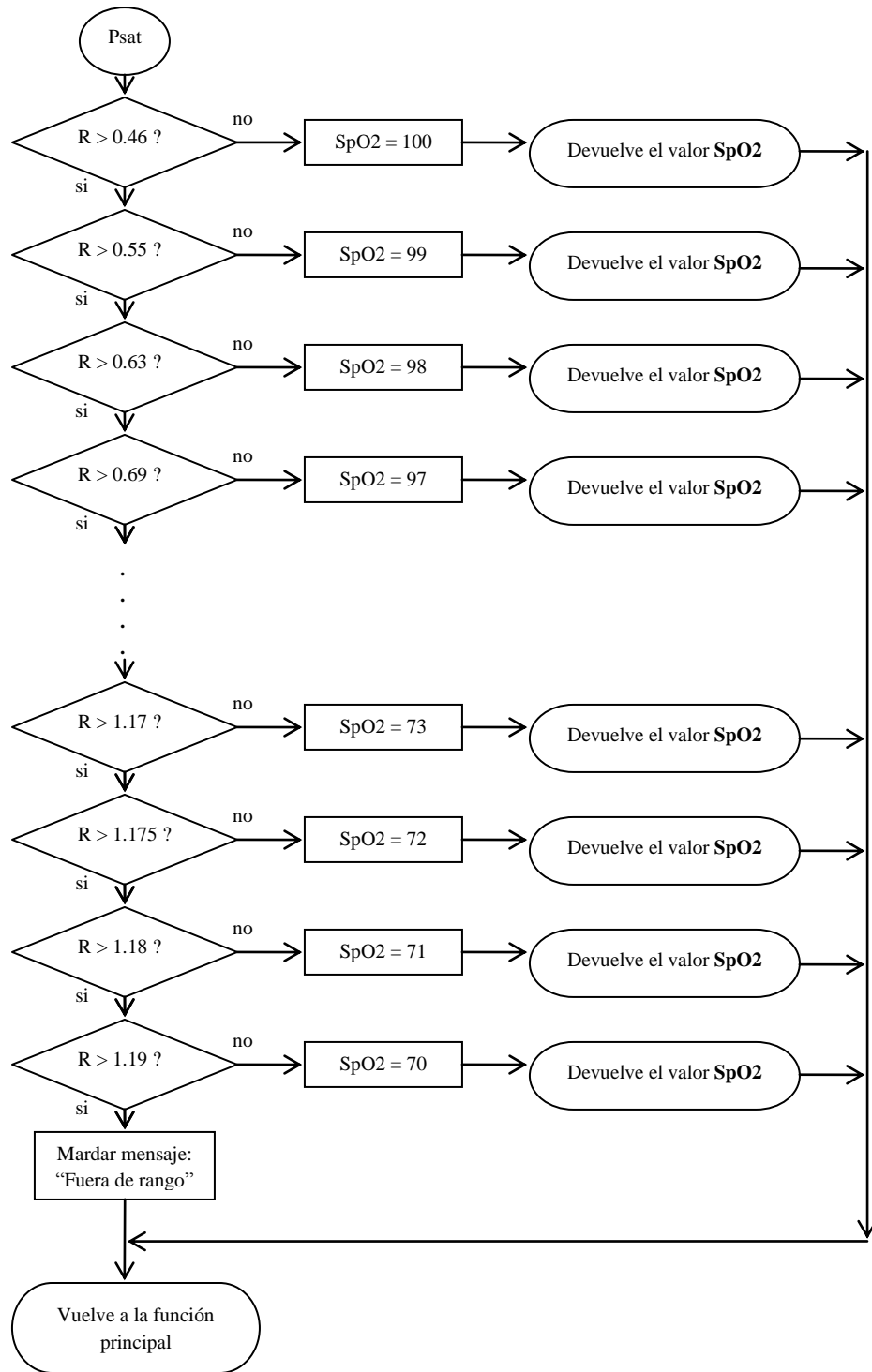




## Diagrama de flujo para calcular la frecuencia cardiaca



## Diagrama de flujo para obtener el %SpO<sub>2</sub> en función R



# DIAGRAMA ESQUEMATICO DEL OXIMETRO DE PULSO

